

### **REMARKS**

Claims 1-9, as amended, are pending in this application. In this Response, Applicants have amended certain claims. In particular, Applicants have amended claim 1 to clarify that the present invention prepares a list of tables that are related to returning the set of results but that are not directly referred to by the SQL statement. The process described by the present invention is used to examine the list to determine which of the related tables can be eliminated from the join.

In light of the Office Action, Applicants believe these amendments serve a useful clarification purpose, independent of patentability. Accordingly, Applicants respectfully submit that the claim amendments do not limit the range of any permissible equivalents. As no new matter has been added, Applicants respectfully request entry of the amendments at this time.

### **THE REJECTIONS UNDER 35 U.S.C. § 102**

At pages 2-6 of the Office Action, the Examiner rejected claims 1-9 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,640,221 to Levine *et al.* ("Levine"). Applicants submit that Levine does not teach the present invention for at least the reasons that follow.

Levine relates to SQL queries and statements in relational databases. *See* Col. 1, lines 8-11. SQL statements that include more than one join operation have two types of result sets: (i) a final result set; and (ii) an intermediate result set. *See* Col. 1, lines 56-58. The final result set is the final table that is generated from the tables being joined after all the join operations are executed. *See* Col. 1, lines 58-60. The intermediate result set is the intermediate table that is generated from just two of the tables (or one table and another intermediate result set) being joined in one of the join operations. *See* Col. 1, lines 60-64. Levine seeks to allow the manipulation of intermediate results sets by providing a system and method for configuring, sequencing, and viewing joins in a SQL query. *See* Abstract.

More specifically, Levine teaches a method of automatically sequencing tables joined in an SQL query. *See* Col. 4, lines 10-12. To begin, Levine provides a join object for each table. *See* Col. 4, lines 12-13. The join object includes a table relationships table that lists related tables. *See* Col. 4, lines 13-14. A first list of tables in an SQL query is then provided. *See* Col. 4, lines 14-15. A first table from the first list is then moved into a second list, and the first table is deleted from the first list. *See* Col. 4, lines 15-17. If the first table (which is now in the second list) has a related table in the table relationships list, the related table is also moved from the first list to the second list,



and the related table is deleted from the first list. *See* Col. 4, lines 17-20. This process is repeated until “there are no tables remaining in the first list.” *See* Col. 4, lines 27-28. Thus, the process disclosed by Levine uses no discretion, *i.e.*, it removes tables from the first list until no tables remain. *Id.*

In this manner, Levine teaches a process by which a SQL join operation is performed on tables in the SQL query as well as all tables related to tables in the SQL query. Levine does not, however, teach or suggest a process for preventing execution of unnecessary joins between tables in a database. Instead, Levine teaches a process that allows all tables related to a table in an SQL query to be included in the SQL join operation. Therefore, Levine discloses two lists: (i) a first list that includes related tables; and (ii) a second list on which the SQL join operation is performed.

In contrast, the present invention relates to a computer implemented method of preventing the execution of unnecessary joins between tables in a database. *See* Amended Claim 1. The method includes presenting a SQL statement to a database. *Id.* The SQL statement has a scope that extends to a set of tables in the database and returns a set of results from the database. *Id.* The present invention includes the step of preparing a list of tables that are related to the set of results but that are not directly referred to by the SQL statement. *Id.* The execution of a join involving any of the tables remaining in the list is prevented. As shown in amended independent claim 1, there is at least one table remaining in the list. Claim 1 has been amended further to clarify only one list, *i.e.*, the list of related tables, is generated by the present invention.

Thus, in one aspect of the present invention there is only one list generated. The list includes tables that are related to returning the set of results but that are not directly referred to by the SQL statement. Tables that are removed from the list are deemed necessary for returning the correct results for the SQL query. At least one table that remains on the list, however, is prevented from being joined.

It appears that the Examiner has maintained the § 102 rejection based on Levine because Levine removes related tables from a first list and moves them to a second list. Because Levine removes tables from the first list until the list is empty, joins are prevented from being executed on tables in the first list (because it is empty). However, the removal of the tables from the first list in Levine is performed without discretion until the first list is empty. In contrast, claim 1 of the present invention has been amended to clarify that one embodiment of the present invention



generates only one list that includes tables that are related to, but not directly referred to, by an SQL query. A related table of the present invention is only included in the SQL query if it is deemed necessary to returning an accurate result. In other words, if a table is deemed necessary, it is removed from the list (*see, e.g.*, the “removing” step of claim 1). If a related table is not deemed necessary to returning an accurate result, the table remains on the list, and it is prevented from being included in a join operation.

Applicants believe that the present amendments to claim 1 clarify that operation of one aspect of the present invention. Levine specifically states that all tables are removed from the first list until no tables remain. Moreover, the removal of tables from the first list in Levine is done without discretion. Therefore, the first list in Levine will always be empty after the process described by Levine is implemented. Thus, Levine does not teach each and every feature of the present invention as recited by amended claim 1, *i.e.*, there is at least one table remaining on the list recited by claim 1 of the present invention.

For at least the reasons set forth above, Applicants submit that claim 1 is in condition for allowance. Applicants further submit that claims 2-9 are also in condition for allowance by virtue of their dependency on claim 1, but also for additional novel features recited therein. As such, reconsideration and allowance of the pending claims is respectfully requested.

### **CONCLUSION**

All claims are believed to be in condition for allowance. If the Examiner believes that the present amendments and remarks still do not resolve all of the issues regarding patentability of the pending claims, Applicants invite the Examiner to contact the undersigned attorneys to discuss any remaining issues.

A Petition for Extension of Time is submitted herewith extending the time for response three months to and including December 27, 2007. No other fees are believed to be due at this time. Should any fee be required, however, please charge such fee to Bingham McCutchen LLP Deposit Account No. 50-4047, Order No. 19111.0126.



Respectfully submitted,  
BINGHAM MCCUTCHEN LLP

Dated: December 27, 2007

By:



Siddhesh W. Pandit, Registration No. 58,572

BINGHAM MCCUTCHEN LLP

2020 K Street, NW

Washington, D.C. 20006

(202) 373-6513 Telephone

(202) 373-6001 Facsimile